오라클11g

ORACLE DB IP 10번대로 변동시

IPCONFIG/ALL

IPCONFIG/RELEASE

IPCONFIG/RENEW

Chapter15 뷰

-View :실제 테이블에 저자오딘 데이터를 뷰를통해서 볼 때

사용목적 : 직접적인 테이블의 접근을 제한하기위해 사용

복잡한 질의를 쉽게 만들기 위해서 사용

특징 : 테이블에 대한 제한을 가지고 일정한 부분만 보일수있는 가상의 테이블

실제로 자료는 갖지않으며, 하나의 테이블에 뷰의 개수는 제한이 없다.

-뷰 생성 :

1. 권한 부여 : GRANT CREATE VIEW TO SCOTT
2. VIEW생성 : 정의한 가져올 테이블에서 보여줄칼럼들을 WHERE조건에 맞춰 VIEW를 생성해준다.

OR REPLACE : 기존에 뷰가 존재하더라도 삭제하지않고 새로운 구조의 뷰로 변경할수있음

FORCE : 기존 테이블의 존재 여부에 상관없이 뷰를 생성한다

WITH CHECK OPTION : 해당 뷰를 통해서 볼수있는 범위 내에서만 UPDATE,INSERT가 가능하다

CREATE [OR REPLACE ,FORCE,WITH CHECK OPTION]VIEW VIEW\_NAME(칼럼별칭 지정)

AS

SELECT 보여줄칼럼들(그룹함수가 들어갈경우 별칭을 지정해줘야한다 SUM(SAL) AS “SALSUM”

FROM 가져올 테이블

WHERE 조건

[WITH CHECK OPTION, WITH READ ONLY 각각 조건절에 사용된 컬럼의 값을 수정못하게,DML 수행이 불가능하게 설정]

-생성한 VIEW에 관환 데이터 딕셔너리

: SELECT VIEW\_NAME,TEXT SELECT TABLE\_NAME,TEXT

FROM USER\_VIEWS FROM USER\_TABLES

--뷰의 종류



-단순뷰에대한 데이터 조작

:단순 뷰에서는 DML명령어(INSERT/UPDATE/DELETE) 문을 사용할수있다

EX)INSERT INTO VIEW\_NAME

VALUES(데이터,데이터,데이터)

:단순 뷰에서 DML명령어를 사용 할수 없는경우

기본테이블의 칼럼이 NOT NULL제약 조건이 지정되어있는경우

가상칼럼이 뷰에 정의되어있는 경우

DISTINCT(중복제거)를 포함한 경우

그룹함수나 GROUP BY절을 포함한 경우

-단순 뷰의 칼럼에 별칭 부여하기

:뷰 생성 내용에 포함되어있음

복합뷰 : 두개 이상의 기본 테이블에 의해 정의한 뷰

CREATE VIEW VEW\_NAME

AS

SELECT 보여줄 칼럼명

FROM 테이블1 E,테이블2 D

WHERE E.조인시킬칼럼=D.조인시킬칼럼

ORDER BY EMPNO DESC(EMPNO를 DESC내림차순으로 정렬)

-뷰 삭제 : 뷰는 실체가 없는 가상 테이블이기 때문에 뷰를 삭제한다는 의미는 USER\_VIEW 데이터 딕셔너리에 저장 되어 있는 뷰의 정의를 삭제하는 것을 의미

SELECT VIEW\_NAME,TEXT

FROM USER\_VIEWS; ->데이터 딕셔너리에 저장되어있는 뷰를 보여줌

DROP VIEW VIEW\_이름->뷰삭제

-ROWNUM칼럼을 이용한 뷰 활용하기

ROWNUM칼럼은 오라클 내부적으로 부여되어 INSERT문에 의해 입력한 순서에 따라 1씩 증가되면서 값이 지정된다

CREATE OR REPLACE VIEW VIEW\_NAME

AS

SELECT 칼럼명

FROM 테이블명

ORDER BY HIREDATE

->

SELECT HIREDATE FROM VIEW\_NAME

WHERE ROWNUM<=5

ORDER BY HIREDATE를 해줬음에도 ROWNUM의 특성상 정렬된 HIREDATE순으로 출력되는 것이 아니라 ROWNUM 초기 데이터 입력순으로 1~5까지 만 출력되게된다

->HIREDATE순으로 1~5까지 출력하기위해선 인라인뷰로 TOP-N을 구할수있다

여기서 인라인 뷰란 서브쿼리의 일종으로 보통 FROM절 내부에 사용된 서브 쿼리문으로 CREATE VIEW를 사용하는 것이 아니라 SQL문 내부에 뷰를 정의하여 이를 테이블 처럼 사용하는 것을 의미

SELECT HIREDATE

FROM(SELECT HIREDATE FROM 테이블명 ORDER BY HIREDATE)

WHERE ROWNUM<=5;

CHAPTER 17 시퀀스,인덱스,동의어

시퀀스(SEQUENCE)란 유일(UNIQUE)한 값을 생성해주는 오라클 객체

시퀀스를 생성하면 기본키와 같이 순차적으로 증가하는 칼럼을 자동적으로 생성가능

SEQUENCE 생성방법

CREATE SEQUENCE SEQUENCE\_NAME

[START WITH N] 시퀀스의 시작값을 지정 DEFAULT 1

[INCREMENT BY N] 시퀀스의 증가값을 설정 DEFAULT1

[MINVALUE N | NOMINVALUE] 시퀀스의 최솟값을 지정 | 시퀀스의 최소값은 무한소

[MAXVALUEW N | NOMAXVALUE] 시퀀스의 최대값을 지정 | 시퀀스의 값을 무한대로 지정

[CYCLE | NOCYCLE] 지정한 시퀀스의 최대값까지 증가가 완료되면 다시 최솟값에서 시작 | 최대값을 넘어가면 오류발생 DEFAULT : NOCYCLE

[CACHE N | NOCACHE] 오라클서버가 미리 지정하고 메모리에 유지할 값의 수 DEFAULT : 2

SEQUENCE 사용하기

1.테이블의 구조를 복사한 테이블생성

-DROP TABLE 테이블명

-CREATE TABLE 테이블명

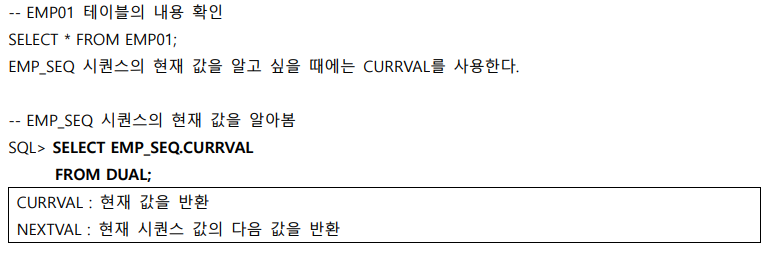
AS SELECT 칼럼명

FROM 복사할 테이블명(시퀀스 저장할테이블,등등)

WHERE 1=0;

2.생성한 시퀀스로부터 번호를 할당받아 데이터를 추가

INSERT INTO 복사한테이블명 VALUES(시퀀스명.NEXTVAL, 등등)



시퀀스 수정

ALTER SEQUENCE SEQUENCE\_NAME

[INCREMENT BY N] 시퀀스의 증가값을 설정 DEFAULT1

[MINVALUE N | NOMINVALUE] 시퀀스의 최솟값을 지정 | 시퀀스의 최소값은 무한소

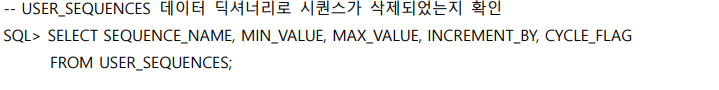
[MAXVALUEW N | NOMAXVALUE] 시퀀스의 최대값을 지정 | 시퀀스의 값을 무한대로 지정

[CYCLE | NOCYCLE] 지정한 시퀀스의 최대값까지 증가가 완료되면 다시 최솟값에서 시작 | 최대값을 넘어가면 오류발생 DEFAULT : NOCYCLE

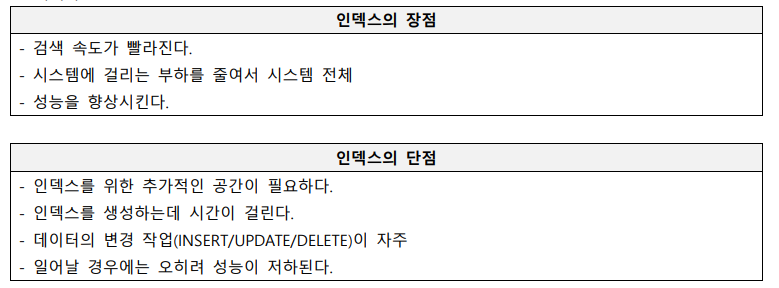
[CACHE N | NOCACHE] 오라클서버가 미리 지정하고 메모리에 유지할 값의 수 DEFAULT : 2

시퀀스 삭제

DROP SEQUENCE 시퀀스이름 PURGE; PURGE하게되면 쓰레기값없이 완전삭제



-인덱스



CREATE TABLE EMP01

AS

SELECT \* FROM EMP 로 테이블을 복사 하게되면 구조와 내용만을 복사 할뿐 제약조건은 복사 되지않는다

-> CREATE INDEX INDEX\_NAME

ON TABLE\_NAME(COLUMN\_NAME) : 테이블의 해당 칼럼에 인덱스네임을 생성

인덱스삭제

DROP INDEX INDEX\_NAME PURGE;

동의어 : 동일한 TABLE에 대해 다른이름으로 대체할수있으며, 다른사용자 소유의 객체에 대한 엑세스를 간단하고 쉽게 할수있으며, 객체의 원래의 이름을 숨길 수 있다.

객체의 소유권은 해당 객체를 생성한 사용자에게 있으므로 다른 사용자가 해당 객체에 접그하기 위해서는 소유자로부터 접근 권한을 부여 받아야한다

CREATE [PUBLIC] SYNONYM SYNONYM\_NAME ->공용동의어: PUBLIC으로 할경우 권한을 주는 사용자가 정의한 동의어로 누구나 사용할수있다 전용 동의어: PUBLIC을 사용하지않고 객체에 대한 접근 권한을 부여받은 사용자가 정의한 동의어로 해당사용자만 사용가능

사용방법

1. 해당 테이블에대한 접근 권환을 다른사용자에게 부여해준다

GRANT SELECT ON 해당테이블 TO SCOTT

1. CREATE [PUBLIC] SYNONYM SYNONYM\_NAME : 이때 SYNONYM\_NAME은 사용할 동의어

FOR 해당테이블 : 이때 PUBLIC으로 만들어주면 해당사용자 말고 권환을 부여해준 사용자에서도 접근이 가능하게해준다

동의어 삭제

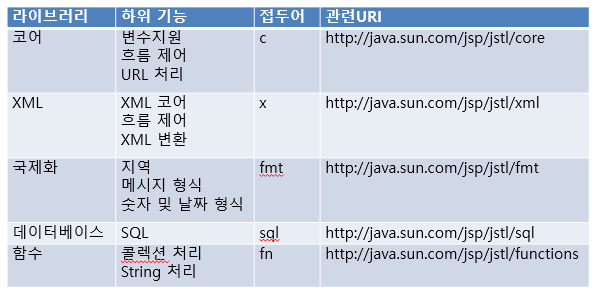
DROP SYNONYM SYNONYM\_NAME PURGE;

JSP381-592PDF P96~p144 자료, 20230414/el&jstl. el&jstl.txt

CHAPTER 16 표준 태그 라이브러리(JSTL)

JSTL 관련 JAR파일 https://maven-repository.dev.java.net/repository/jstl/jars/

JSTL의 종류

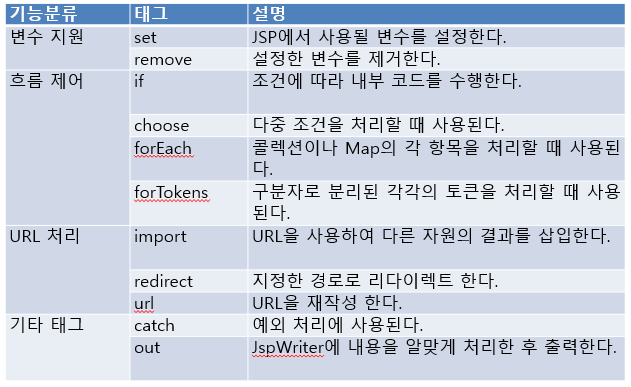


해당 라이브러리를 사용할 때 접두어와 관련 URI를 사용하여 taglib디렉티브를 추가해주어야한다

->코어라이브러리 taglib디렉티브추가

<%@ taglib prefix=”c” uri=<http://java.sun.com/jsp/jstl/core> %> 로 taglib을 추가시켜야 jstl 코어 라이브 러리를 사용할수있다

-코어태그 라이브러리



-변수 지원태그

<c:set>태그 사용하여 EL변수 설정하기

1. <c:set var=”varName” value=”varValue”[scope=”영역”]/> == pageContext.setAttribute(varName, varValue, scope)를 호출하여 el변수의 이름이 varName에 값이 varValue인 el변수를 설정해준다

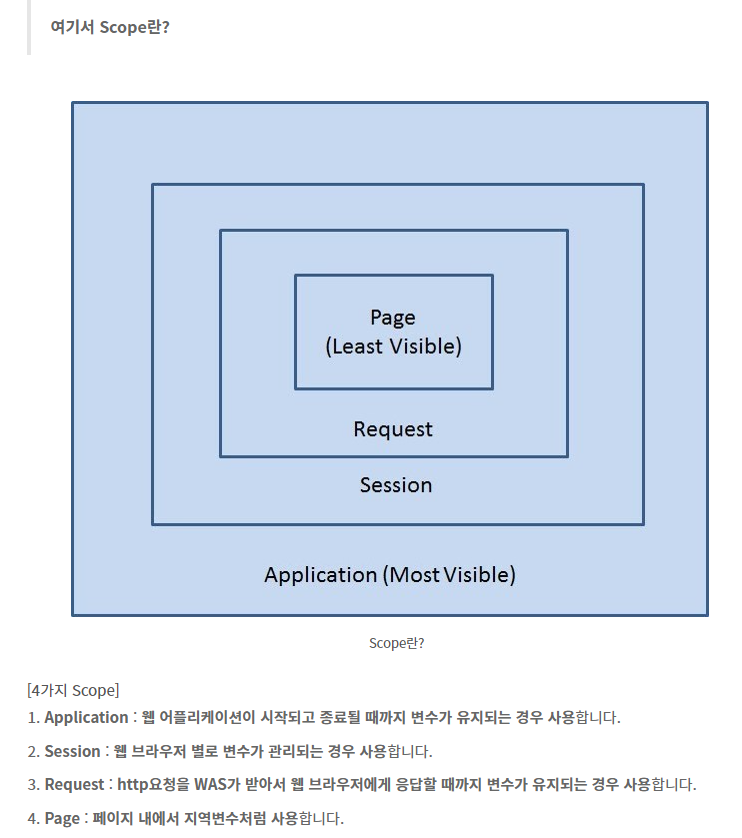
Or

<c:set var=”varName” [scope=”영역”]>varValue</c:set>

var: 값을 지정할 EL변수의 이름

value : 변수의 값을 지정하며 표현식,EL,정적인 텍스트를 사용 하여 값을 지정

scope : 변수를 저장할 영역을 지정 값은 page, request, session, application중 하나가 온다 기본값은 page



2.<c:set>로 객체의 프로퍼티 값을 설정

<c:set target=”대상” property=”프로퍼티이름” value=”값” />

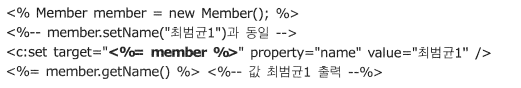
<c:set target=”대상” property=”프로퍼티이름”> value </c:set>

- target : 프로퍼티 값을 설정할 대상 객체를 <%= 변수 %>나 el변수 ${varName}으로 지정한다

이때 대상객체는 자바빈 객체나 MAP이여야한다

-property : 설정할 프로퍼티의 이름 으로 자바빈 객체인 경우 set메소드를 제공해줘야하고 map인경우 map.put(프로퍼티이름, 값)을 이용해서 값을 추가한다

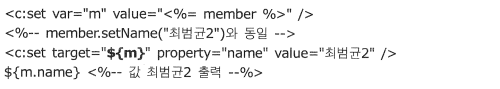
-value : 설정할 프로퍼티의 값



1.member 객체의생성

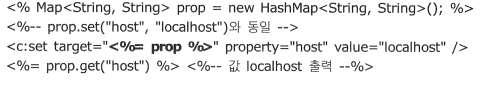
3. 2를해주는거 동일하며 target에는 목표객체가 property 와 value에는 각각의 속성 값을 설정해준다

4.3에서 객체에 속성값을 설정을 해주었기 때문에 getName()을 할수있다



1.el변수의 이름이 m 에 값이 member(위에서 생성한 Member객체)인 el변수를 생성

3.m인 el변수에->Member의 객체이기때문에 name에 값이 최범균2인 프로퍼티를 설정한다



1.hashmap객체 prop를 생성

3.변수 prop를 target으로 프로퍼티 host에 localhost를 넣어준다

<c:set var=”pr2p” value=”${prop}”/>해주고

<c:set target=”${pr2p}” property=”host” value=”localhost”/>

<c:set var=”favorithost” value = “#{pr2p.host}”/>

${favorithost} 해주면 localhost가 출력된다 or <%= pr2p.get(“host”)%> or ${pr2p.get(“host”)}

-<c:remove var=”name”/> name인 el변수를 삭제한다

-<c:if>태그

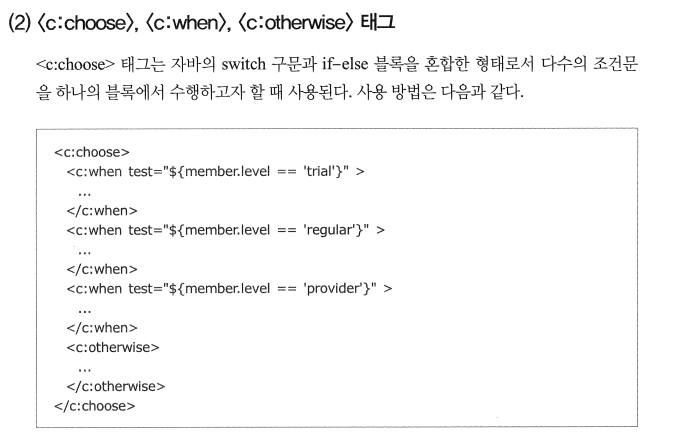
<c:if test=”조건”>

… //”조건”이 true일경우 …내용을 실행한다

</c:if>

-<c:choose>,<c:when>,<c:otherwise>태그

<c:choose>태그는 <c:when>태그의 속성값이 true일경우 해당 <c:when>태그의 몸체내용만을 실행하고 모든 <c:when>태그의 test값이 false라면 <c:otherwise>태그가 실행된다 <c:otherwise>태그작성은 필수는아니며 필요한 경우에만 사용



-<c:forEach>태그

<c:forEach>태그는 배열,Collection 또는 Map에 저장되어 있는 값들을 순차적으로 처리할 때 사용할수있는 태그로 java의 for,do-while등을 대신해서 사용할수있다

<